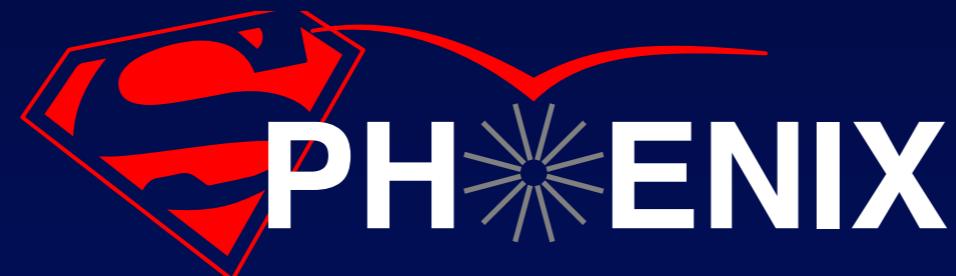


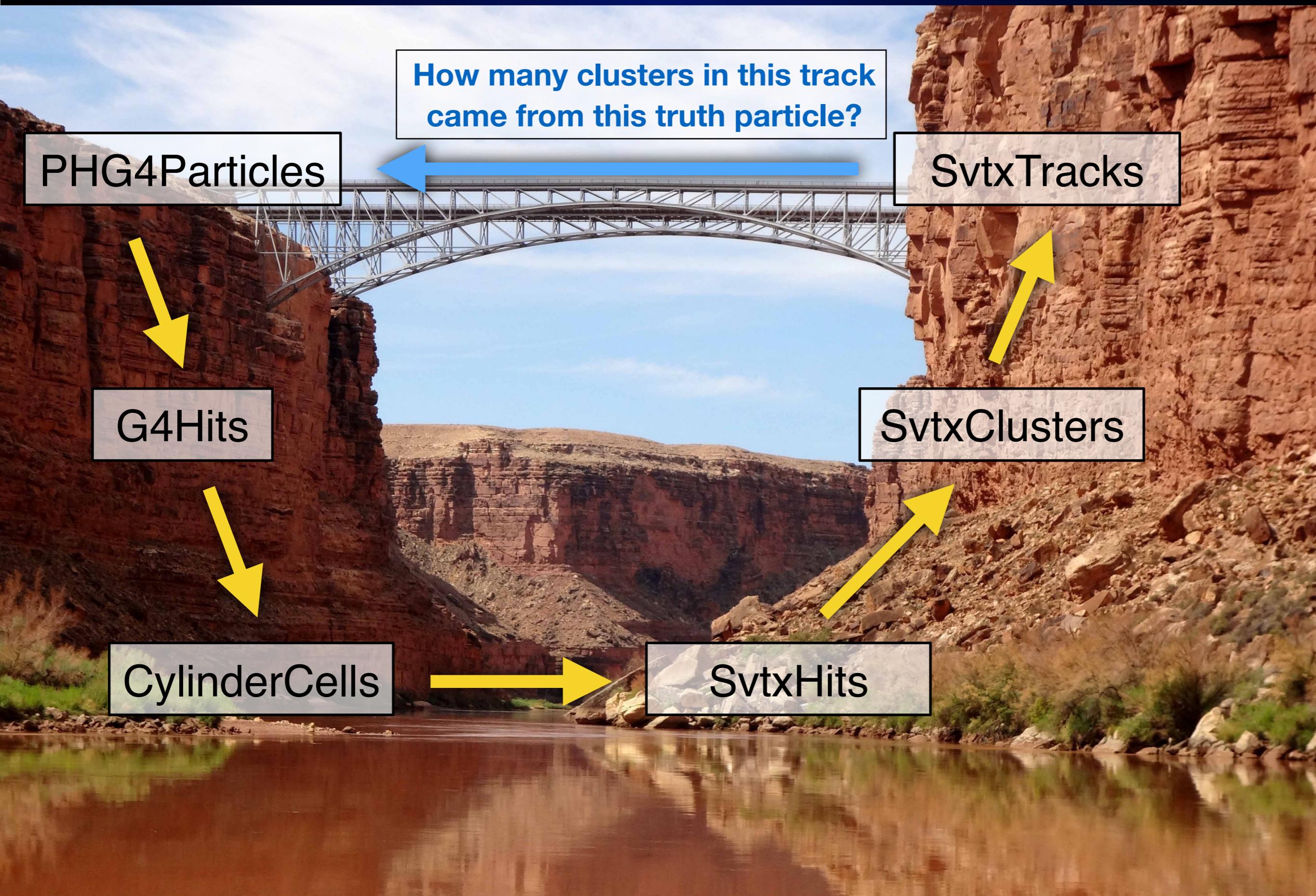
sPHENIX Simulation Evaluators

Michael P. McCumber
Los Alamos National Laboratory

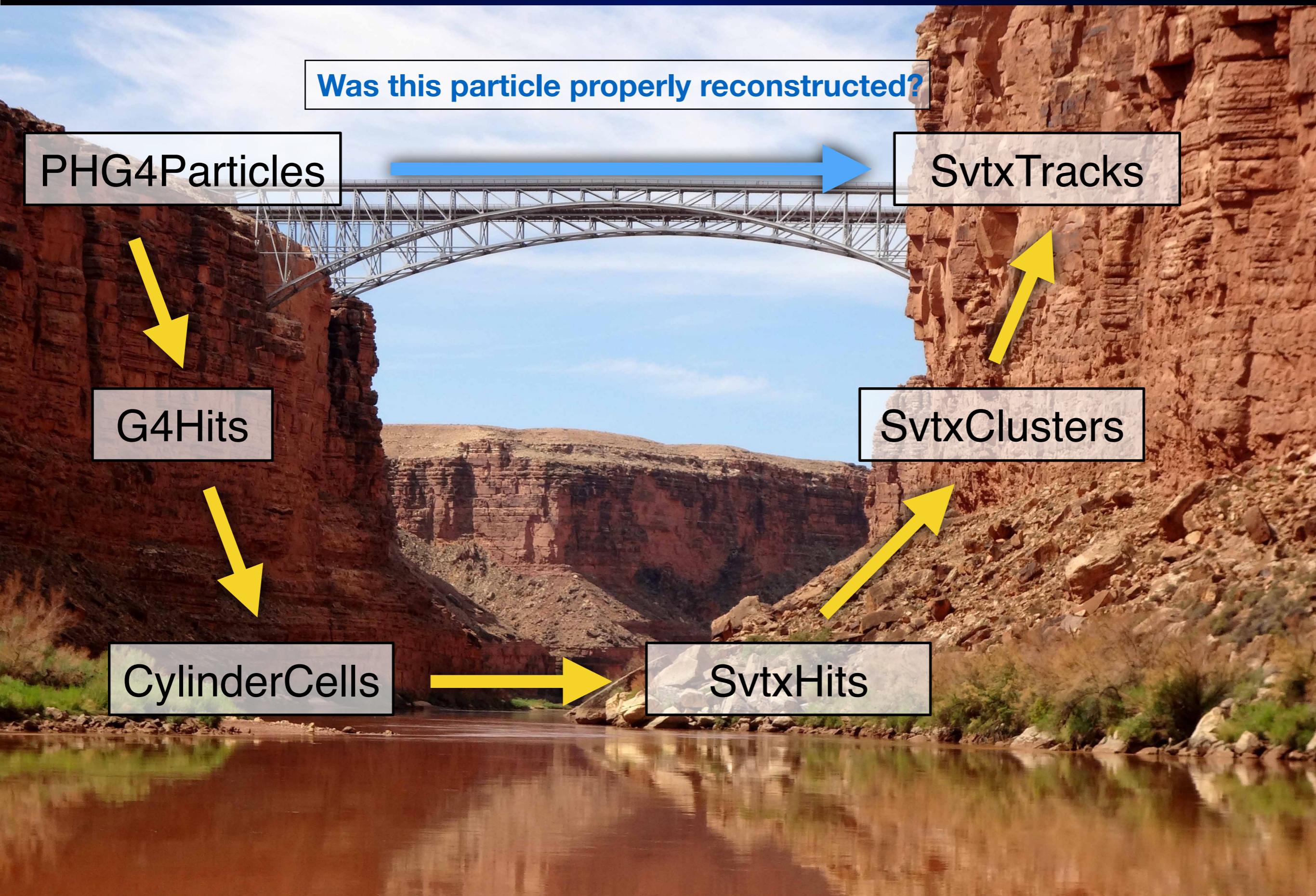


Simulation Meeting
June 23rd 2015

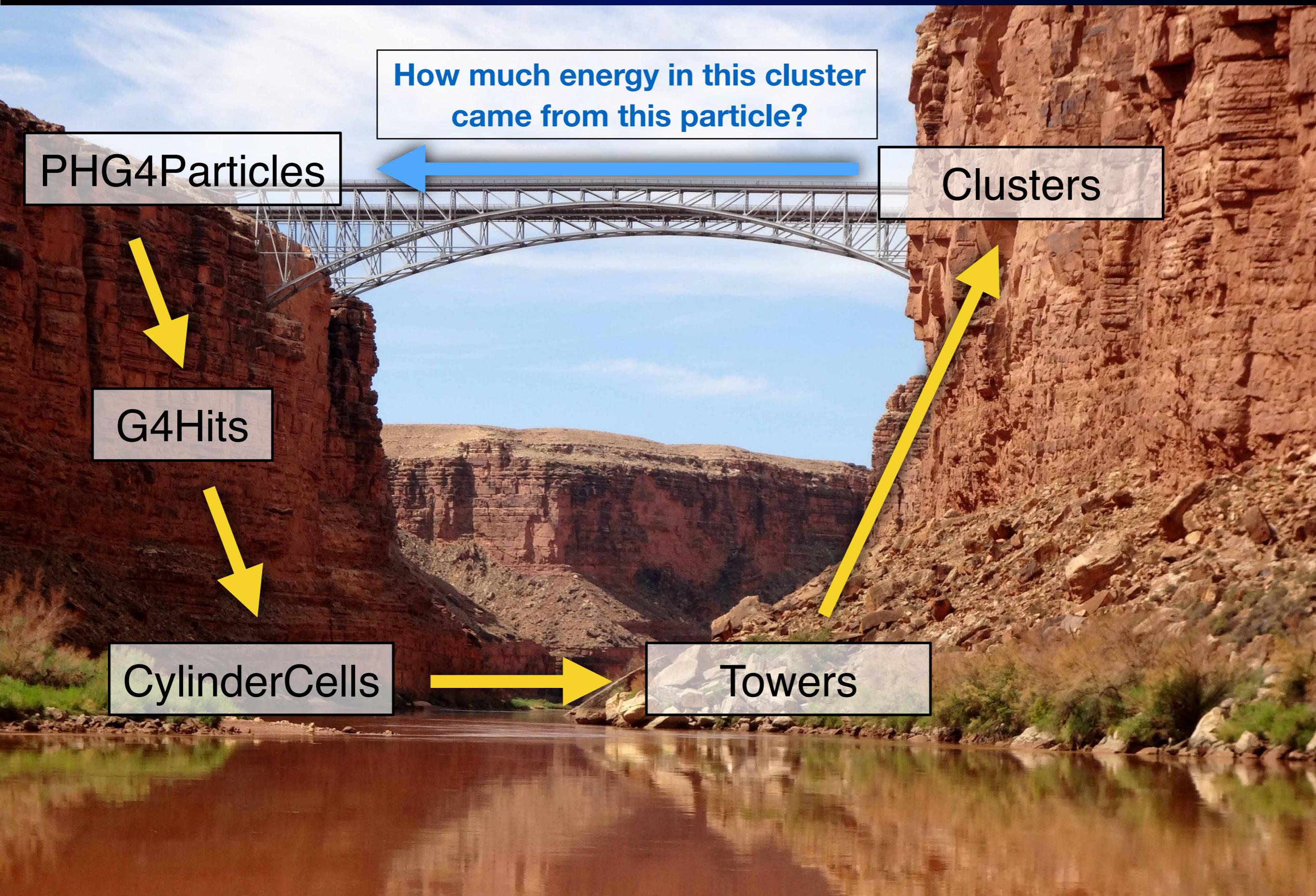
Tracking Evaluator Example



Tracking Evaluator Example



Calorimeter Evaluator Example



Existing Evaluator Codes

g4eval/PHG4Evaluator.{h,C}: Tracking evaluator. Builds the connections between truth particles and reco'd tracks. Crudely ported from SvxEvaluator (the VTX evaluator) which contained optimization maps to prevent storage looping. Outputs: ntuples in a root file for easy plotting.

Tracking Plan (70% completed): Create DST storage for the lookup associations. *Split the evaluator into two modules: one to make the DST objects and another to read them and create the ntuple files. Remove internal optimization maps from Evaluator.*

Usage defined in G4_Svtx.C::Svtx_Eval(string outputfilename).

g4eval/PHG4CalEvaluator.{h,C}: Calorimeter evaluator. Also needs DST objects, separation into two modules, and general house-keeping.

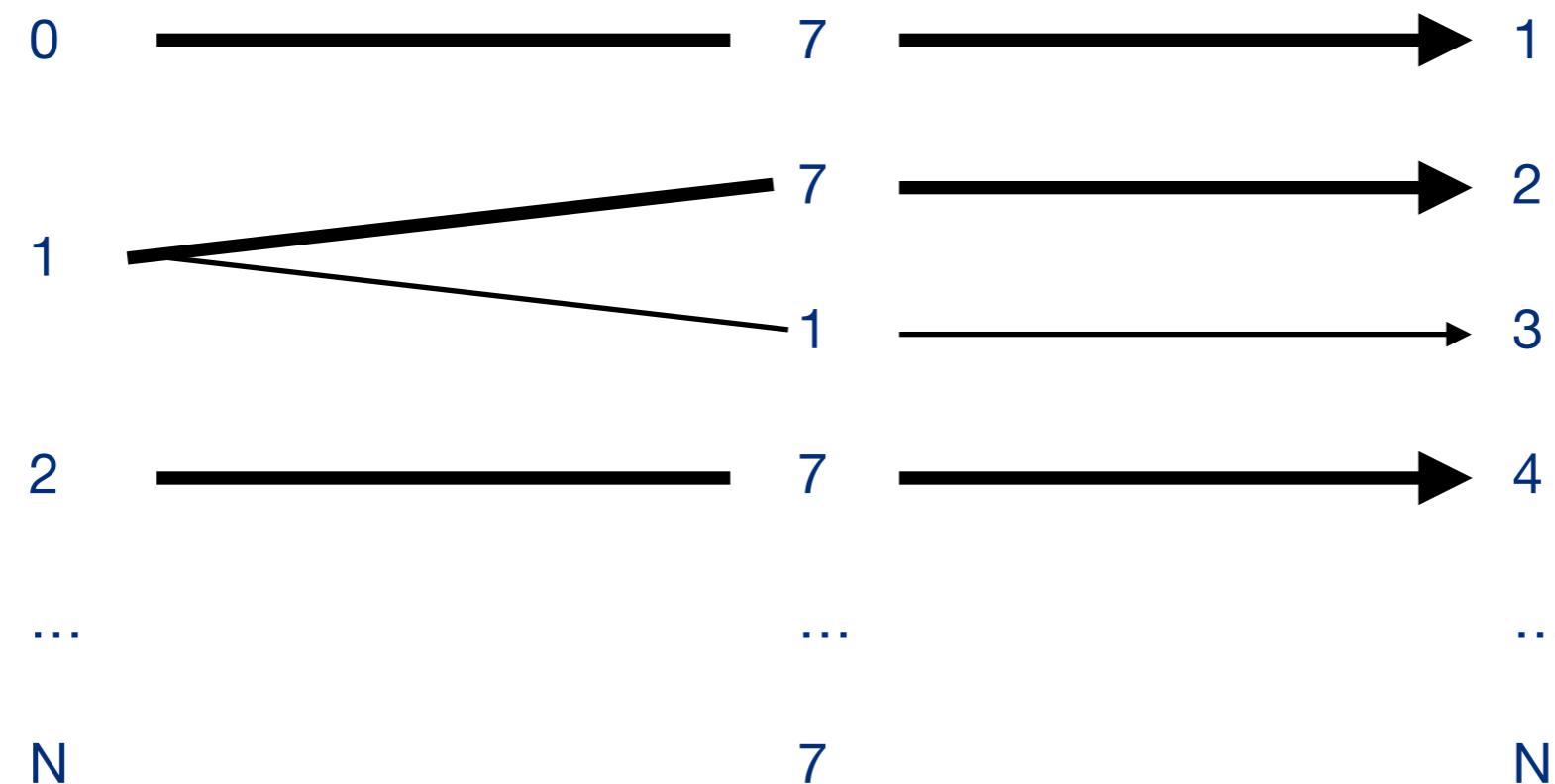
Calorimeter Plan (50% completed): Reuse storage objects, yay!

Usage defined in G4_CEmc_Spacal.C::CEMC_Eval(string outputfilename).

New DST Storage Objects

New objects are **g4eval/EvalLinks{,V1}.{h,C}** use a weighted graph-like design or weighted bi-direction map.

Left Container Object IDs (e.g. Tracks)	Weight Value (e.g. Purity)	Right Container Objects (e.g. Particles)
--	-------------------------------	---



New DST Storage Objects

New objects are **g4eval/EvalLinks{,V1}.{h,C}** use a weighted graph-like design or weighted bi-direction map.

Interface:

```
void set_names(const std::string &left_name,
               const std::string &right_name,
               const std::string &weight_name);
void link(unsigned int left_id, unsigned int right_id, float weight);
void unlink(unsigned int left_id, unsigned int right_id);
void unlink_subleading();
void clear();
```

```
// status
size_t size() const;
std::string get_name_left() const {return _left_name;}
std::string get_name_right() const {return _right_name;}
std::string get_name_weight() const {return _weight_name;}
bool has_link(unsigned int left_id, unsigned int right_id) const;
float get_weight(unsigned int left_id, unsigned int right_id) const;
```

```
// status for missing links
bool is_null_id(unsigned int id) const {return (id == NULLID);}
unsigned int get_null_id() const {return NULLID;}
```

```
// access all associations
std::set<unsigned int> left(unsigned int right_id) const;
std::set<unsigned int> right(unsigned int left_id) const;
```

```
// access best weight association
unsigned int max_left(unsigned int right_id) const;
unsigned int max_right(unsigned int left_id) const;
```

Link creation

(e.g. svtxtrack id, truth id, hit purity)

Get all object ids “right” of a “left id”
(e.g. get all particles contributing
to a particular track)

Get leading object ids “right” of a “left id”
(e.g. get leading truth particle
to a particular track)

New DST Storage Objects

New objects are **g4eval/EvalLinks{,V1}.{h,C}** use a weighted graph-like design or weighted bi-direction map.

Storage: Actually stored in ROOT file

```
std::string _left_name; //< left object container names (e.g. SvtxTrackMap)
std::string _right_name; //< right object container names (e.g. G4TruthInfo)
std::string _weight_name; //< connection weight meaning (e.g. nhits)

/// storage for (left id,right id) => weight value
std::map<std::pair<unsigned int,unsigned int>, float> _links;
```

Lookup Optimization: Refilled on first lookup request

```
#ifndef __CINT__ // hide from dictionary generation
    mutable bool _stale; //!< exclude from ROOT I/O
    mutable std::multimap<unsigned int,unsigned int> _left_right mmap; //!< exclude from ROOT I/O
    mutable std::multimap<unsigned int,unsigned int> _right_left mmap; //!< exclude from ROOT I/O
    mutable std::map<unsigned int,unsigned int> _left_right_map; //!< exclude from ROOT I/O
    mutable std::map<unsigned int,unsigned int> _right_left_map; //!< exclude from ROOT I/O
#endif // __CINT__
```

On the DST

List of Nodes in Fun4AllServer:

Node Tree under TopNode TOP
 TOP (PHCompositeNode)/
 DST (PHCompositeNode)/
 PHG4INEVENT (PHDataNode)
 G4HIT_HCALOUT (PHIODataNode)
 G4HIT_ABSORBER_HCALOUT (PHIODataNode)
 G4HIT_SVTX (PHIODataNode)
 G4HIT_SVTXSUPPORT (PHIODataNode)
 G4HIT_CEMC (PHIODataNode)
 G4HIT_HCALIN (PHIODataNode)
 G4HIT_ABSORBER_HCALIN (PHIODataNode)
 G4HIT_HCALIN_SPT (PHIODataNode)
 G4HIT_MAGNET_0 (PHIODataNode)
 G4HIT_BH_1 (PHIODataNode)
 G4TruthInfo (PHIODataNode)
 PHHepMCGenEvent (PHIODataNode)
 G4CELL_SVTX (PHIODataNode)
 SVTX (PHCompositeNode)/
 SvtxHitMap (PHIODataNode)
 SvtxClusterMap (PHIODataNode)
 SvtxTrackMap (PHIODataNode)
 SvtxVertexMap (PHIODataNode)

Ancestry Lookup:

SvtxClusterMap, G4Hit_SVTX, edep

SvtxClusterMap, G4Hit_SILICON_TRACKER, edep

SvtxTrackMap, G4TruthInfo, purity (clusters with hit contributions)

Forward Lookup:

G4TruthInfo_SvtxTrackMap, nclusters left

SVTX_EVAL (PHCompositeNode)/

SvtxClusterMap_G4HIT_SVTX_Links (PHIODataNode)

SvtxTrackMap_G4TruthInfo_Links (PHIODataNode)

G4TruthInfo_SvtxTrackMap_Links (PHIODataNode)

RUN (PHCompositeNode)/
 CYLINDERGEOM_SVTX (PHIODataNode)
 CYLINDERGEOM_SVTXSUPPORT (PHIODataNode)
 CYLINDERGEOM_HCALIN_SPT (PHIODataNode)
 CYLINDERGEOM_MAGNET_0 (PHIODataNode)
 CYLINDERGEOM_BH_1 (PHIODataNode)
 CYLINDERGEOM_CEMC (PHIODataNode)
 CYLINDERGEOM_HCALIN (PHIODataNode)
 CYLINDERGEOM_HCALOUT (PHIODataNode)
 CYLINDERCELLGEOM_SVTX (PHIODataNode)
 PAR (PHCompositeNode)/
 SVTX (PHCompositeNode)/
 SvtxBeamSpot (PHIODataNode)

DST Usage

SvtxTrackMap_TruthInfo_Links:

example loop over all truth particles associated with a track

```
std::set<unsigned int> truth_ids = _track_truthinfo_links->right(track_id);
for (std::set<unsigned int>::iterator truth_iter = truth_ids.begin();
     truth_iter != truth_ids.end();
     ++truth_iter) {
    unsigned int truth_id = *truth_iter;
    (go fetch truth particle from container)
}
```